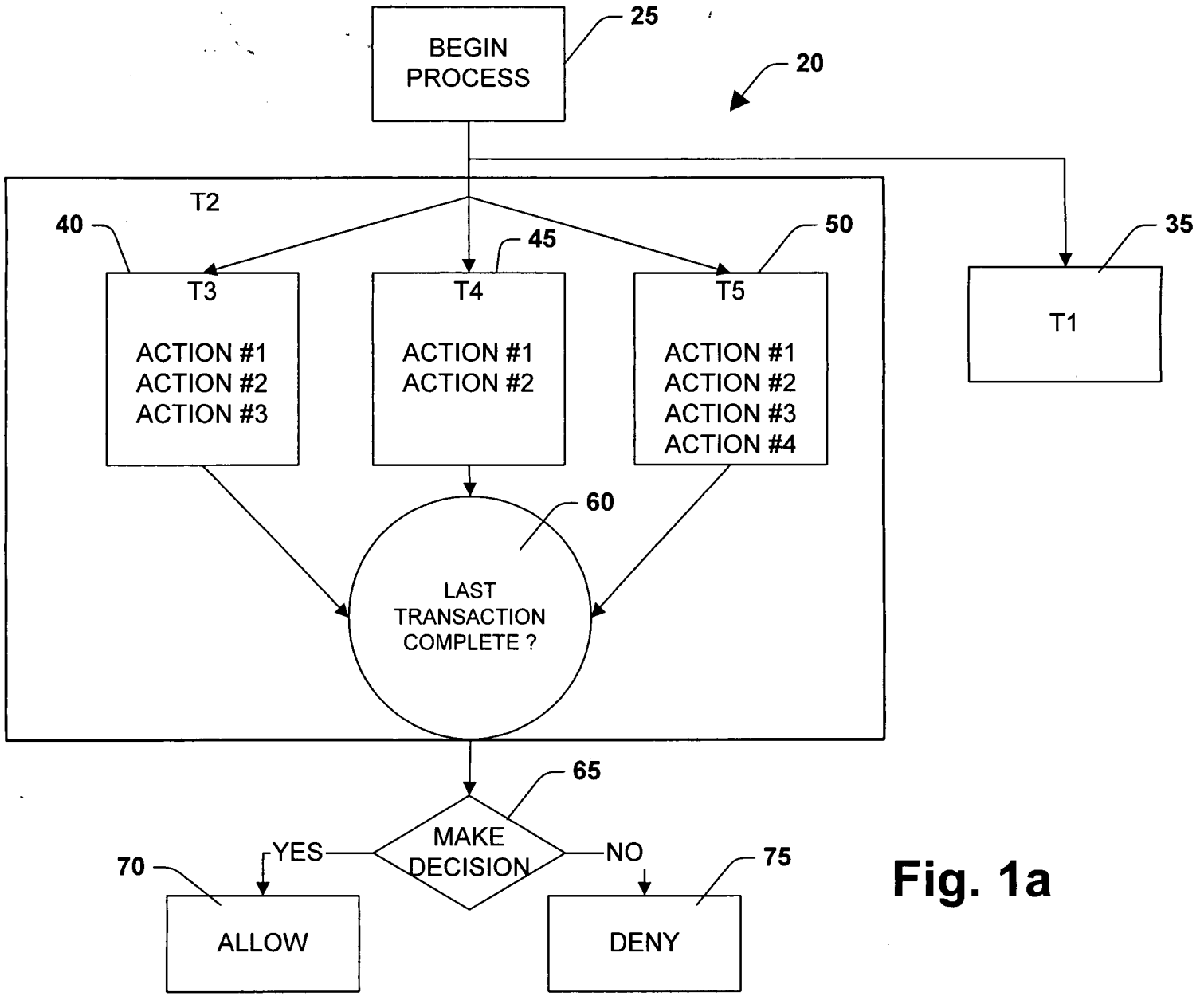

Fig. 1a

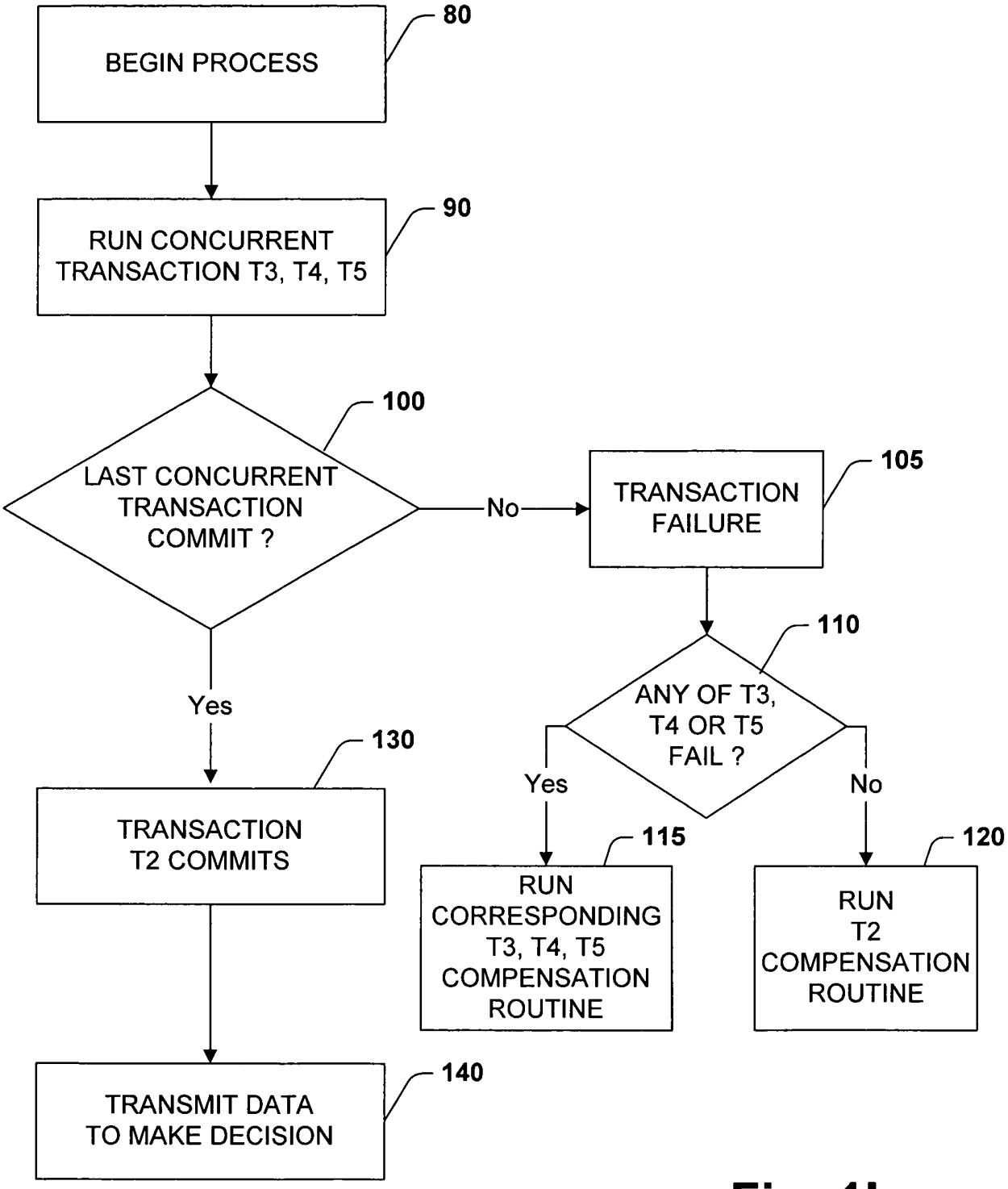

Fig. 1b

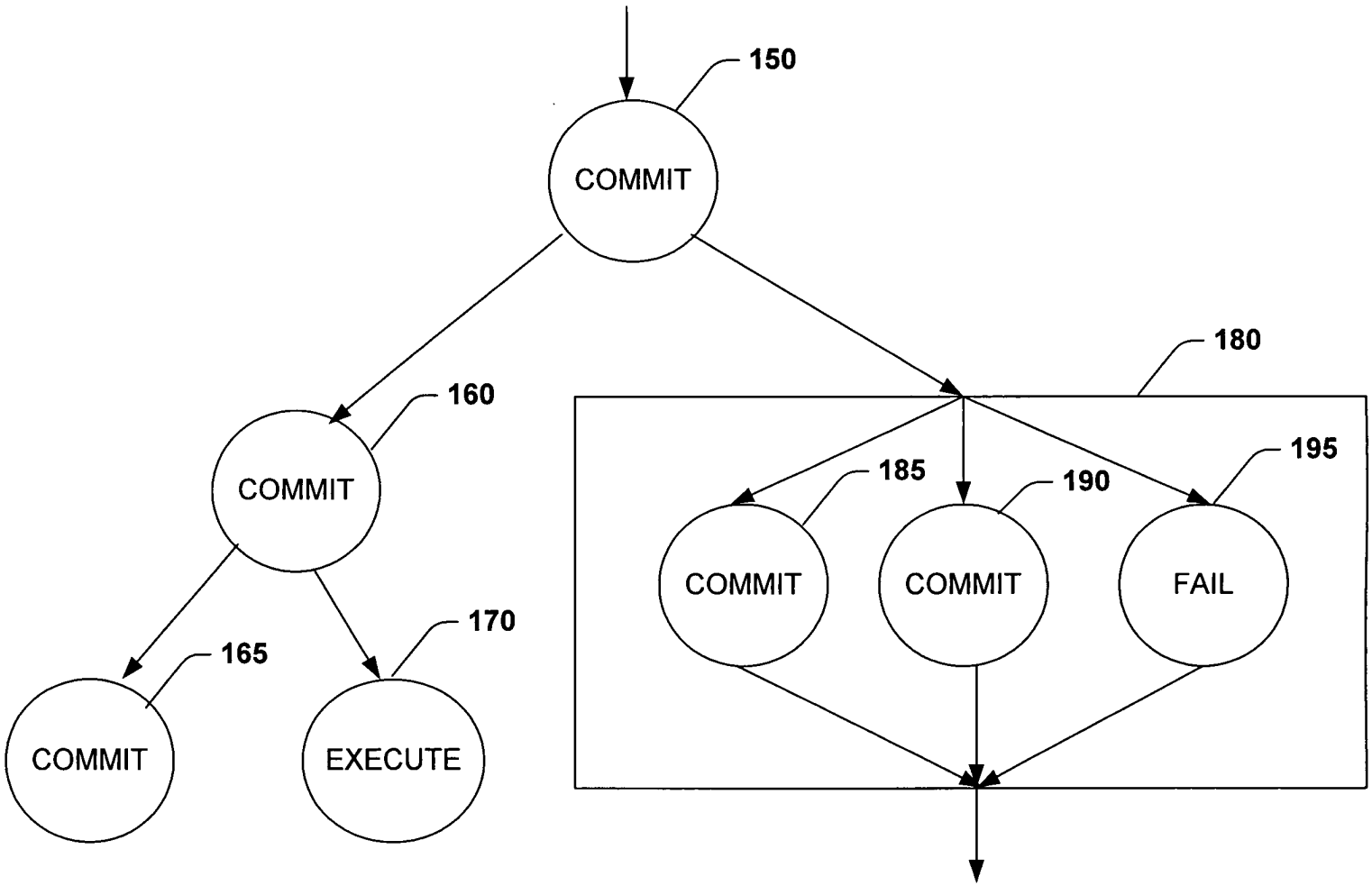

Fig. 1c

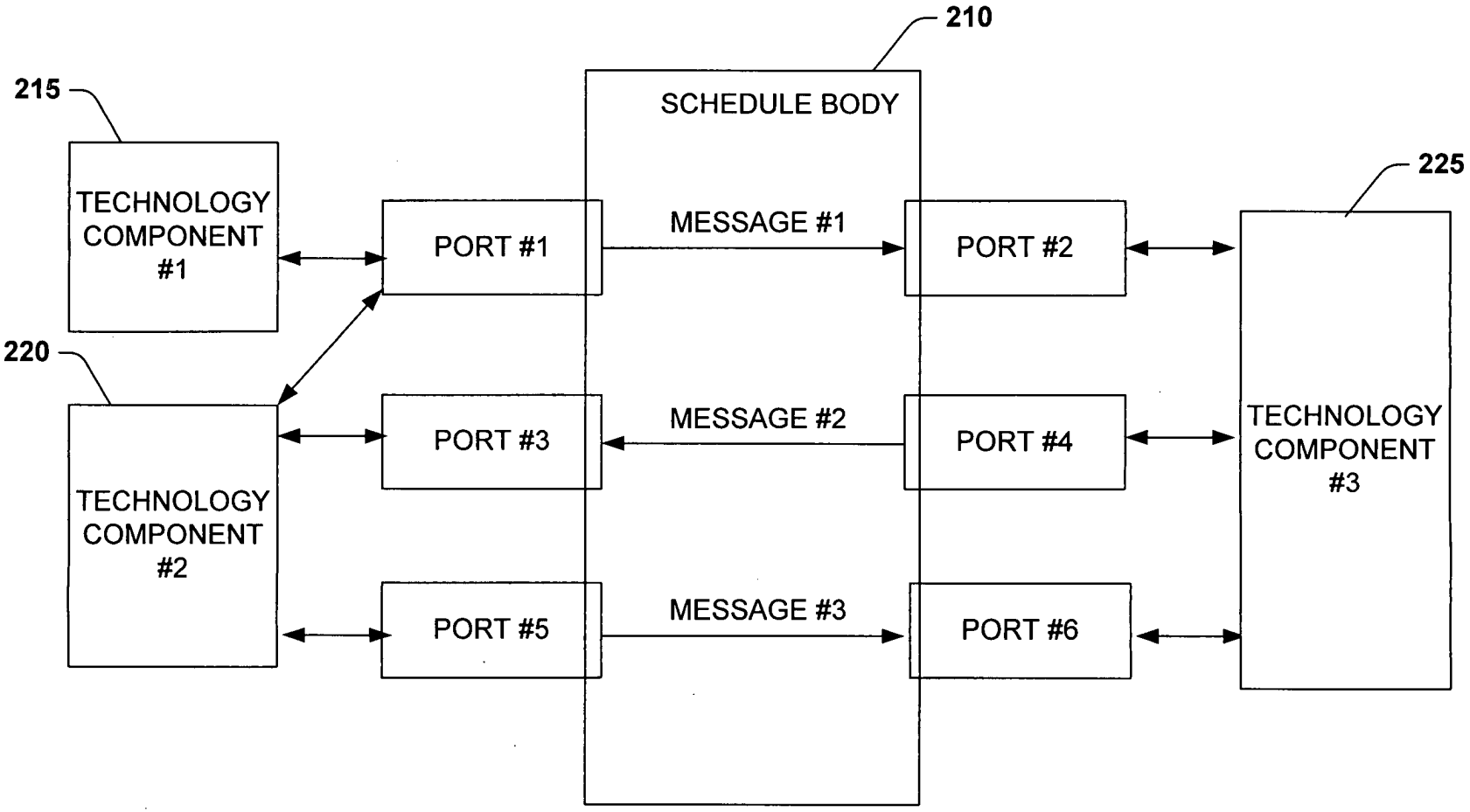

Fig. 1d

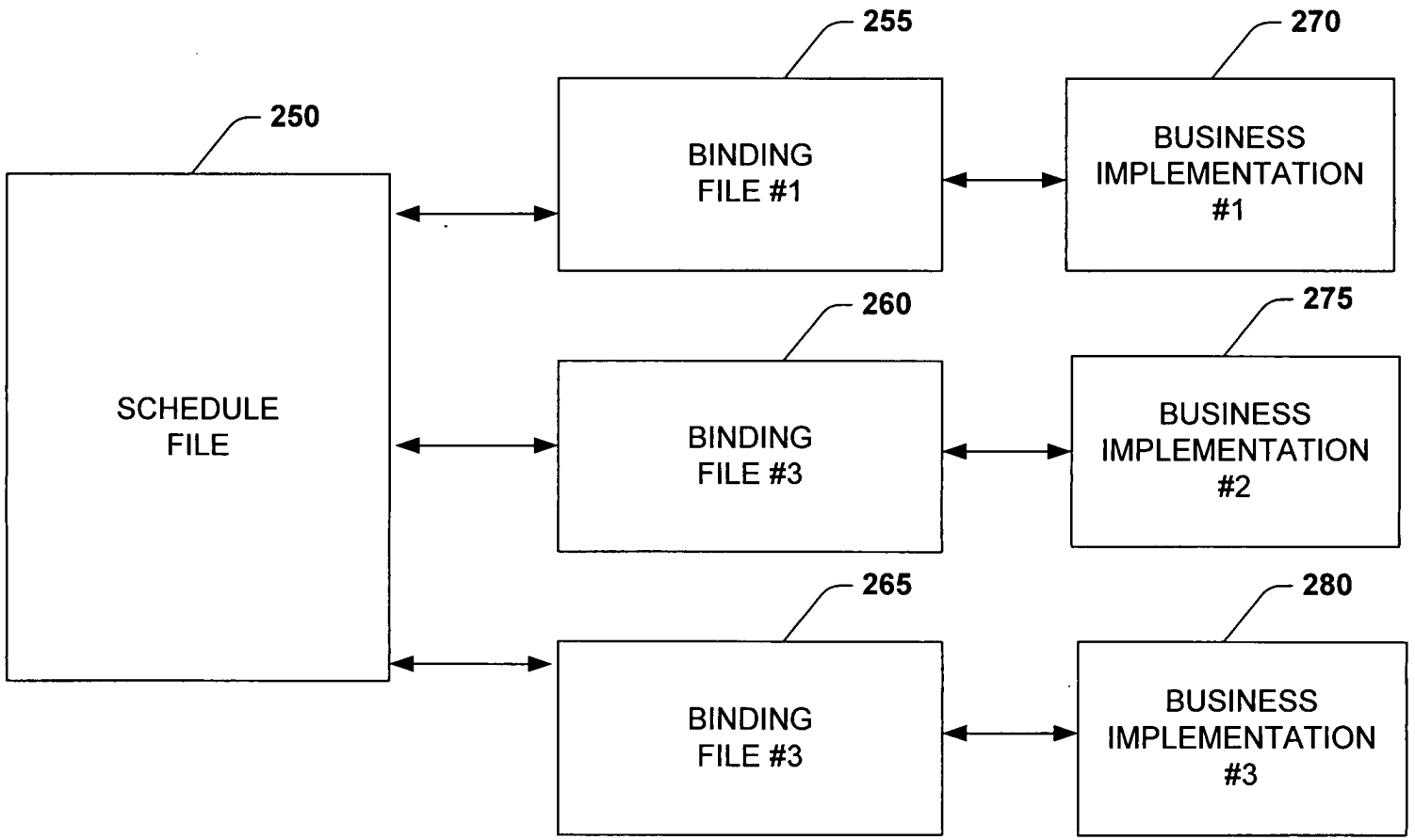

Fig. 1e

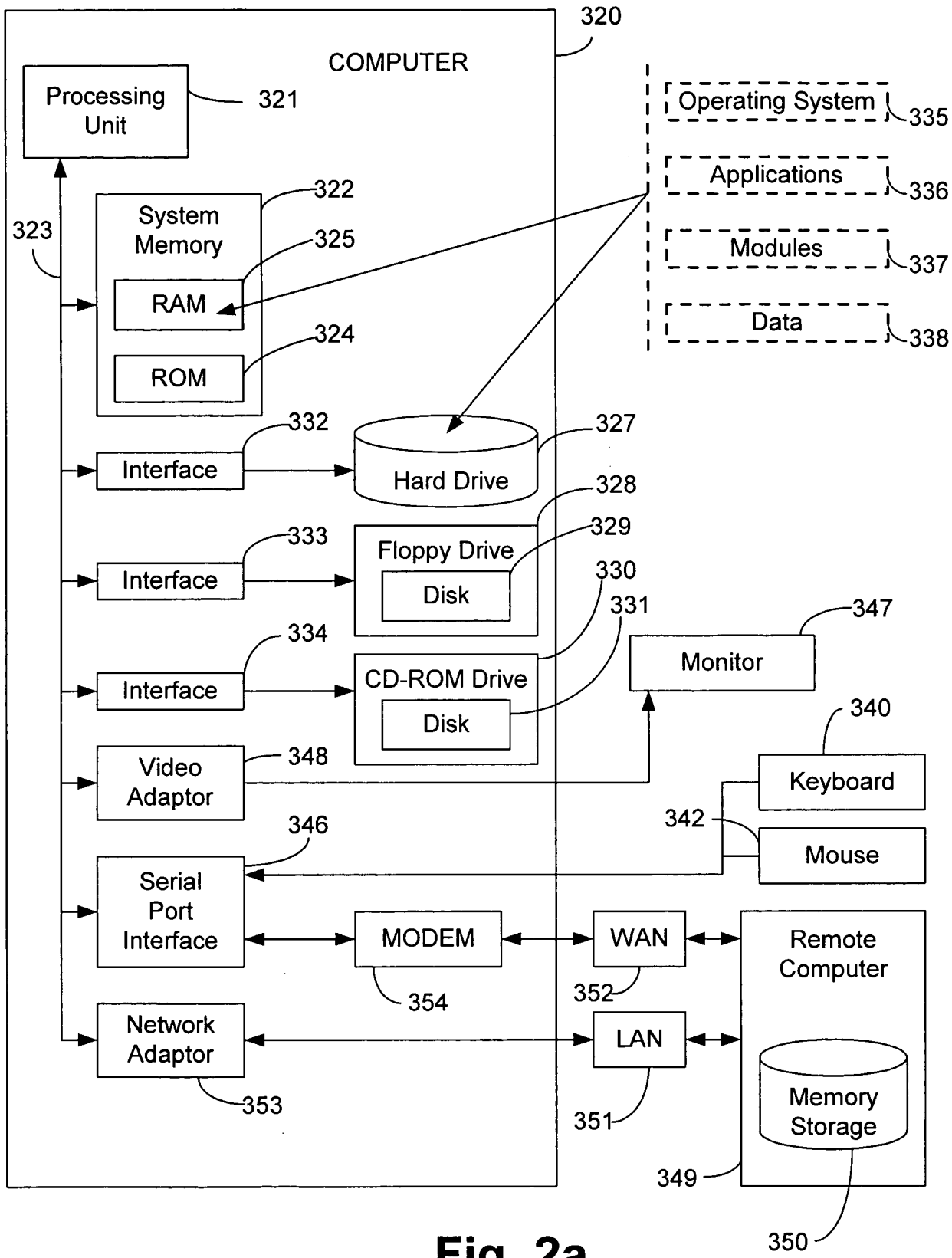
COMPUTER
320
Processing Unit
321
System Memory
322
323
325
RAM
ROM
324
Operating System
335
Applications
336
Modules
337
Data
338
Interface
332
Hard Drive
327
Interface
333
Floppy Drive
328
329
Disk
Interface
334
CD-ROM Drive
330
331
Disk
Monitor
347
Video Adaptor
348
Keyboard
340
342
Mouse
Serial Port Interface
346
MODEM
354
WAN
352
Remote Computer
Network Adaptor
353
LAN
351
Memory Storage
349
350

Fig. 2a

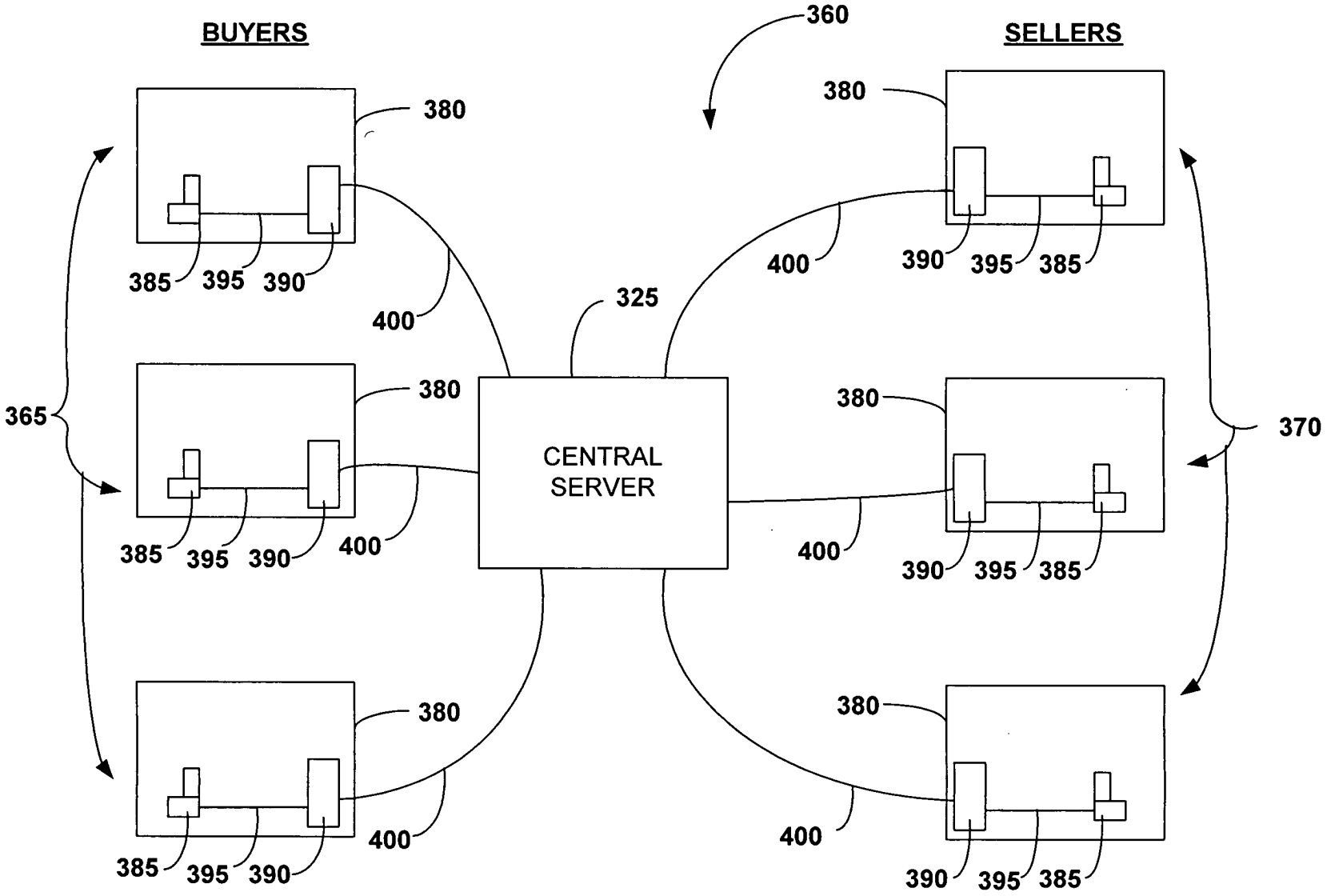

Fig. 2b

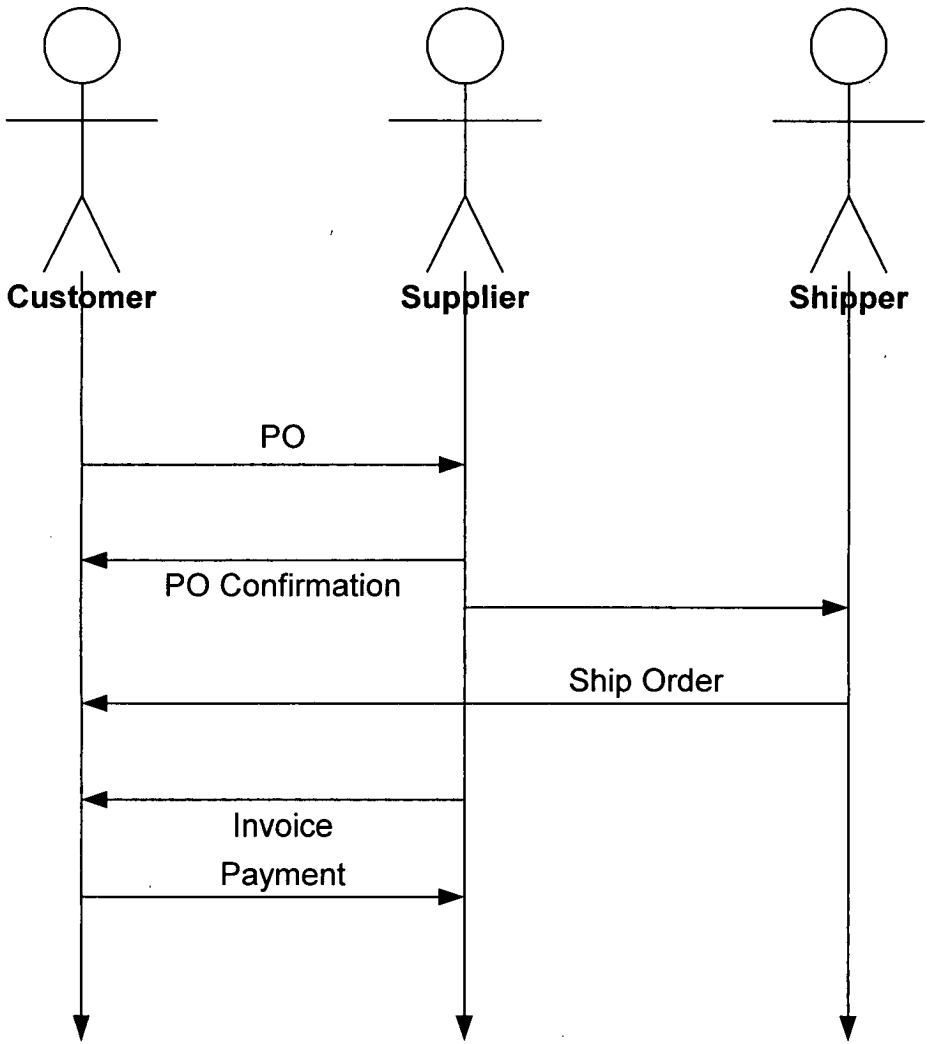

Fig. 3

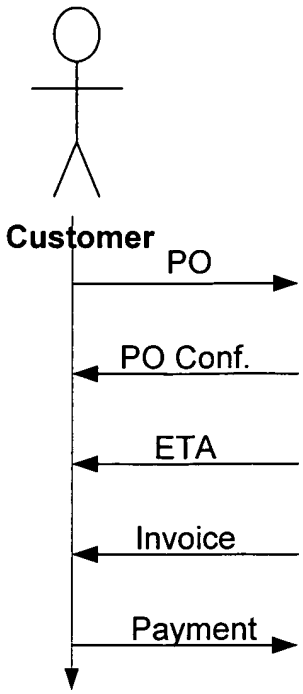

Fig. 4a

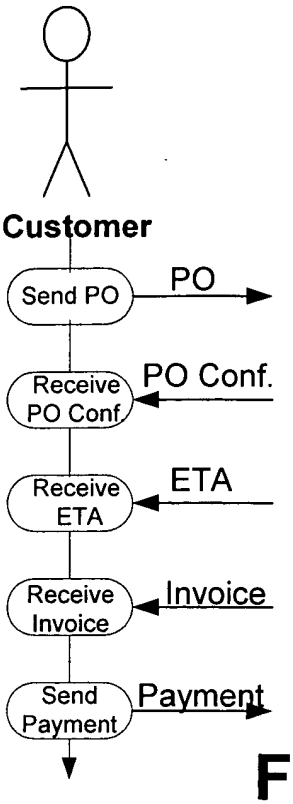

Fig. 4b

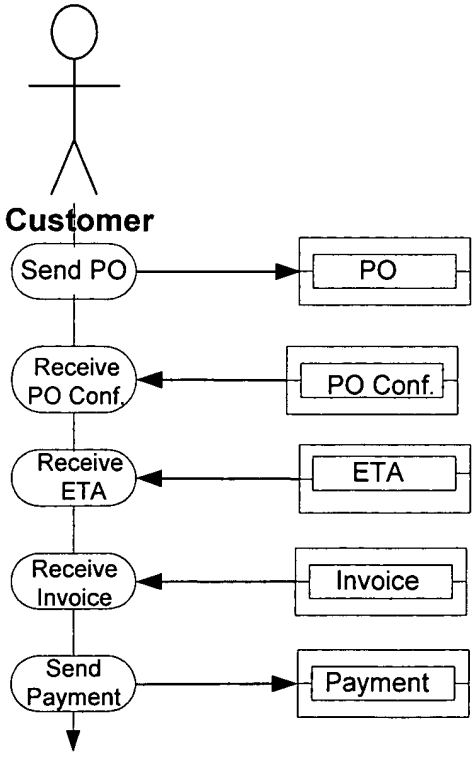

Fig. 4c

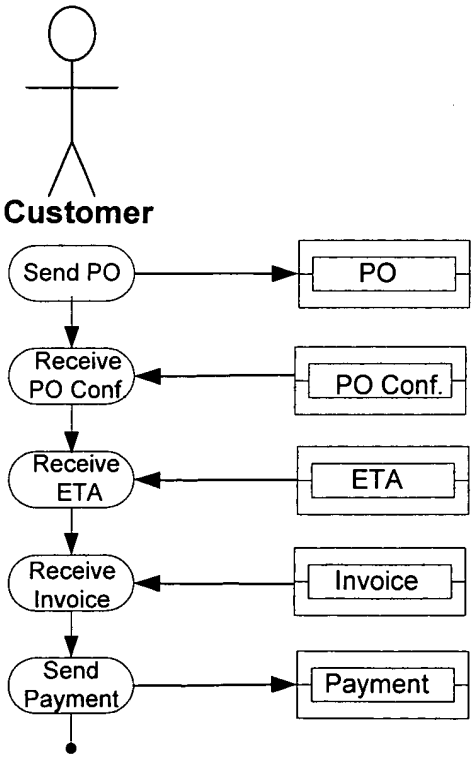

Fig. 4d

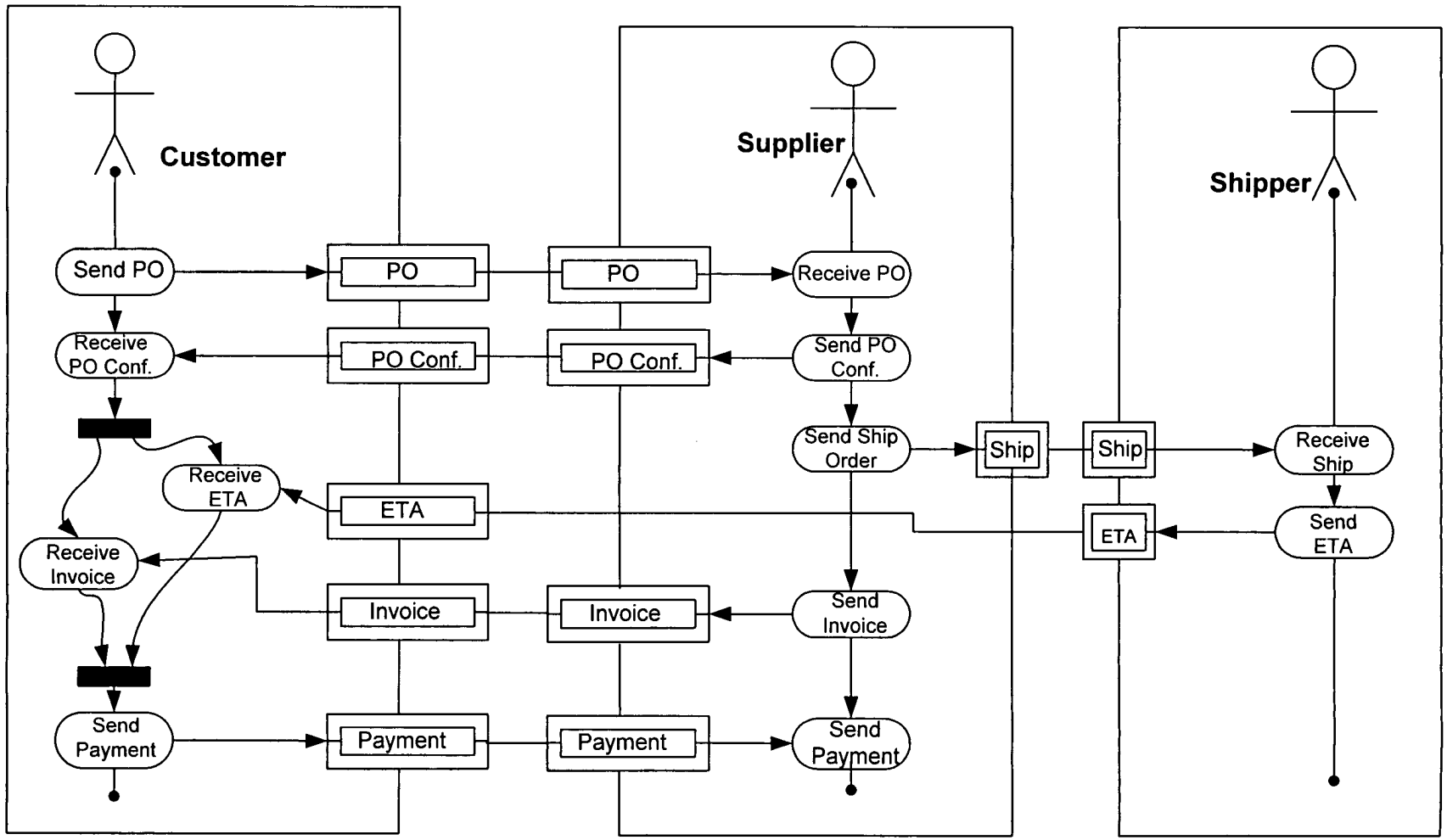

Fig. 5

SLANG

schedule	: : =	header? process? contextRef?
header	: : =	portList? messageList? contextList?
process	: : =	zero | sequence | switch | map | copy| partition | connect | cut
portList	: : =	port*
messageList	: : =	message*
contextList	: : =	context*
zero	: : =	zero
sequence	: : =	genericAction* process? contextRef?
genericAction	: : =	silence | action | task | call | return | release
silence	: : =	silence
action	: : =	source | sink
source	: : =	portRef messageRef contextRef?
sink	: : =	portRef messageRef contextRerf?
task	: : =	action* contextRef?
call	: : =	schedRef portRef* messageRef* contextRef?
switch	: : =	branch* (default process) ? contextRef?
branch	: : =	case process
case	: : =	ruleRef msgRef msgRef
map	: : =	process assignmentList? contextRef?
assignmentList	: : =	asignment*
assignment	: : =	messageRef portRef
copy	: : =	process contextRef?
partition	: : =	process* contextRef?
connect	: : =	process process connectionList contextRef?
connectionList	: : =	connection*
connection	: : =	portRef portRef
cut	: : =	process process process contextRef?

Fig. 6

Schedule (EBNF)

schedule	: : =	*schedule* name identity header process contextRef?
name	: : =	*name identifier*
identity	: : =	*guid GUID*
header	: : =	portList messageList contextList?
portList	: : =	*portList* port*
messageList	: : =	*messageList* message*
contextList	: : =	*contextList* context*
scheduleRef	: : =	*scheduleRef* URI

Fig. 7a

Schedule (XML)

```
< ! ELEMENT schedule (header?, (zero | sequence | switch | map
   | copy | partition | connect | cut) ?, contextRef?)>
< ! ATTLIST schedule
   name        ID          #IMPLIED
   guid        CDATA       #IMPLIED>

< ! ELEMENT scheduleRef EMPTY>
< ! ATTLIST  scheculeRef
   location CDATA #REQUIRED>

< ! ELEMENT header (portList?, messageList?, contextList?)>

< ! ELEMENT portList (port*)>

< ! ELEMENT messageList (message*)>

<! ELEMENT contextList (context*)>
```

Fig. 7b

Example

```
<schedule name="mySchedule">
<header>
   <portList>
      <port name="p0">
      <port name="p1">
   </portList>
   <messageList>
      <message name="m0"/>
      <message name="m1"/>
   </messageList>
</header>
< ! - - The schedule body goes here - - >
</schedule>
```

Fig. 7c

Port (EBNF)		
port	: : =	*port* portName
portName	: : =	*identifier*
portRef	: : =	*portRef URI*

Fig. 8a

Port (XML)

```
< ! ELEMENT port EMPTY>
< ! ATTLIST   port
  name          ID    #REQUIRED>

< ! ELEMENT portRef EMPTY>
< ! ATTLIST   portREF
  location CDATA #REQUIRED>
```

Fig. 8b

Message (EBNF)		
message	: : =	*message* messageName
messageName	: : =	*identifier*
messageRef	: : =	*messageRef URI*

Fig. 9a

Message (XML)

```
< ! ELEMENT  message  EMPTY>
< ! ATTLIST    message
   name        ID        #REQUIRED>

< ! ELEMENT  messageRef   EMPTY>
< ! ATTLIST    messageRef
   location   CDATA    #REQUIRED>
```

Fig. 9b

Context (EBNF)

context	: : =	*context* contextName transactional? compensated? errorCondition?
contextName	: : =	*identifier*
transactional	: : =	*transactional*
compensated	: : =	*compensated* process?
error Condition	: : =	*ruleRef messageRef*

Fig. 10a

Context (XML)

```
< ! ELEMENT  context  (transactional?)>
< ! ATTLIST    context
   name          ID       #REQUIRED>

< ! ELEMENT  transactional   (zero | sequence | switch | map
   | copy | partition | connect | cut) ?>

< ! ELEMENT  contextRef   EMPTY>
< ! ATTLIST    contextRef
   location      CDATA   #REQUIRED>
```

Fig. 10b

Action (EBNF)	
action	: : = source sink
source	: : = *source* portRef messageRef contextRef?
sink	: : = *sink* portRef messageRef contextRef?

Fig. 11a

Action (XML)	
< ! ELEMENT sink	(portRef, messageRef, contextRef?)>
< ! ELEMENT source	(portRef, messageRef, contextRef?)>

Fig. 11b

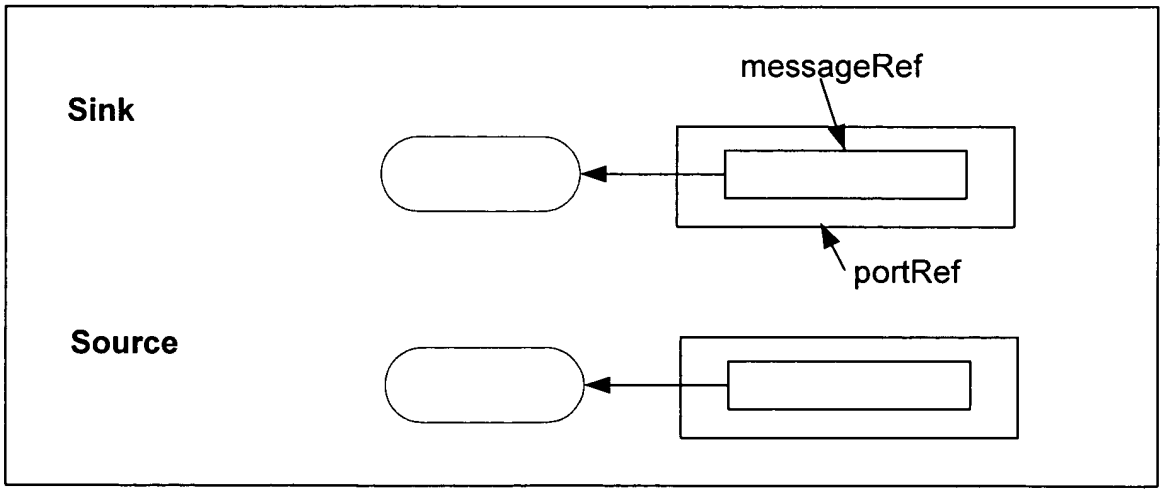

Fig. 11c

Process	
process	: : = zero | sequence | switch | map | copy | partition | connect | cut

Fig. 12

Zero (EBNF)
zero : : = zero

Fig. 13a

Zero (XML)
< ! ELEMENT zero EMPTY>

Fig. 13b

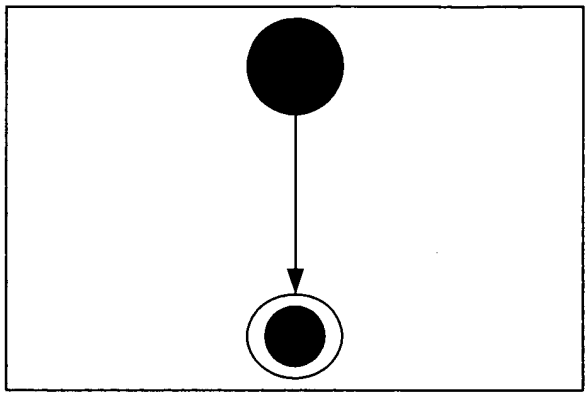

Fig. 13c

Sequence (EBNF)		
sequence	: : =	genericAction+ process? contextRef?
genericAction	: : =	silence | action | task | call | return | relese

Fig. 14a

Sequence (XML)
< ! ELEMENT sequence ((silence | sink | source | task | call | return | release)*, (zero | sequence | switch | map | copy | partition | connect | cut) ?)> < ! ATTLIST sequence ctxt IDREF #IMPLIED>

Fig. 14b

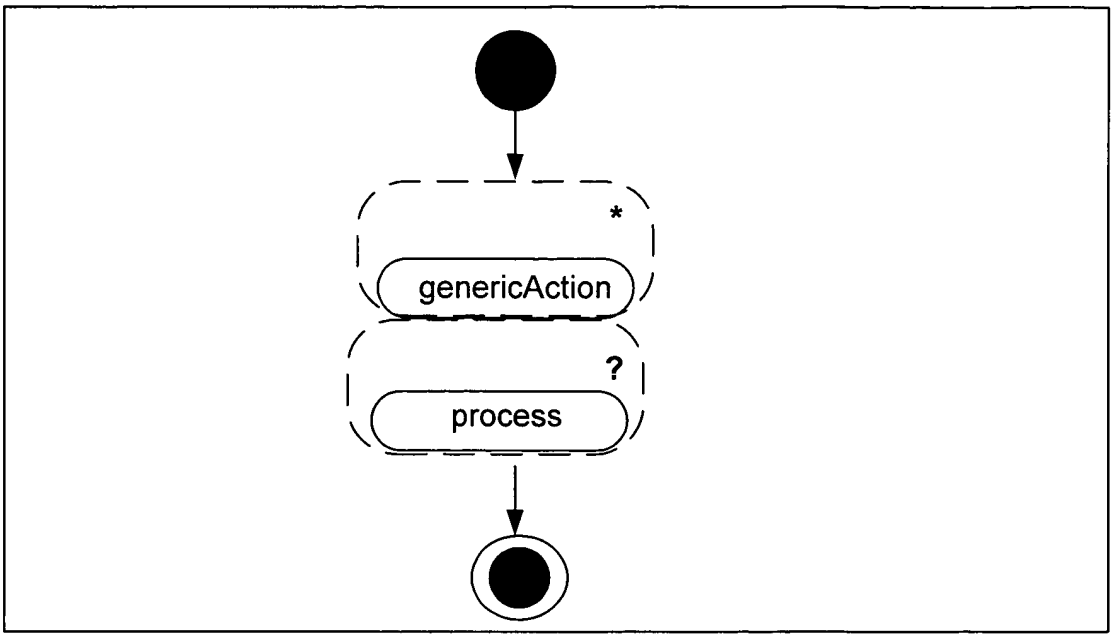

Fig. 14c

Example
<sequence> <sink> <portRef location="p0"/> <messageRef location="m0"/> </sink> <source> <portRef location="p1"/> <messageREf location="m1"/> </source> </sequence>

Fig. 14d

Silence (EBNF)
silence : : = *zero*

Fig. 15a

Silence (XML)
< ! ELEMENT silence EMPTY>

Fig. 15b

Task (EBNF)	
task	: : = action* choice? ctxtRef?
choice	: : = *all* | *any*

Fig. 16a

Task (XML)
< ! ELEMENT task ((sink | source)*, contextRef?)>
< ! ATTLIST task
choice (all / any) "all")

Fig. 16b

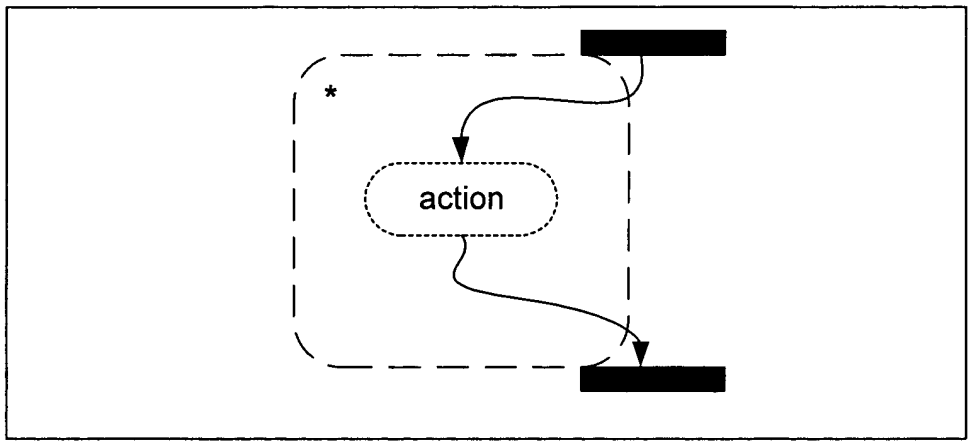

Fig. 16c

Example
<task>
<source>
<portRef location="p0"/>
<messageRef location="m0"/>
</source>
<source>
<portRef location="p1"/>
<message location="m1"/>
</source>
</task>

Fig. 16d

Call (EBNF)
call : : = schedRef portRef* messageRef* contextTef?

Fig. 17a

Call (XML)
< ! ELEMENT call (scheduleRef, portRef*, messageRef*, contextRef)>

Fig. 17b

Return (EBNF)
return : : = *return* contextRef?

Fig. 18a

Return (XML)
< ! ELEMENT return (contextRef?)>

Fig. 18b

Release (EBNF)
release : : = release contextRef?

Fig. 19a

Release (XML)
< ! ELEMENT release (contextRef?)>

Fig. 19b

Switch (EBNF)		
switch	: : =	branch* default? contextRef?
branch	: : =	case process contextRef?
case	: : =	*case* ruleRef messageRef messageRef
ruleRef	: : =	*ruleRef URI*
default	: : =	*default* process

Fig. 20a

Switch (XML)

```
< ! ELEMENT switch (branch* default? contextRef?)>

< ! ELEMENT branch (case, (zero | sequence | switch |
   map | copy | partition | connect | cut), contextRef?)>

< ! ELEMENT case (ruleRef, messageRef, messageRef)>

< ! ELEMENT ruleRef EMPTY>
< ! ATTLIST   ruleRef
   location CDATA #REQUIRED>

< ! ELEMENT default (zero | sequence | switch | map
   | copy | partition | connect | cut), contextRef?)>
```

Fig. 20b

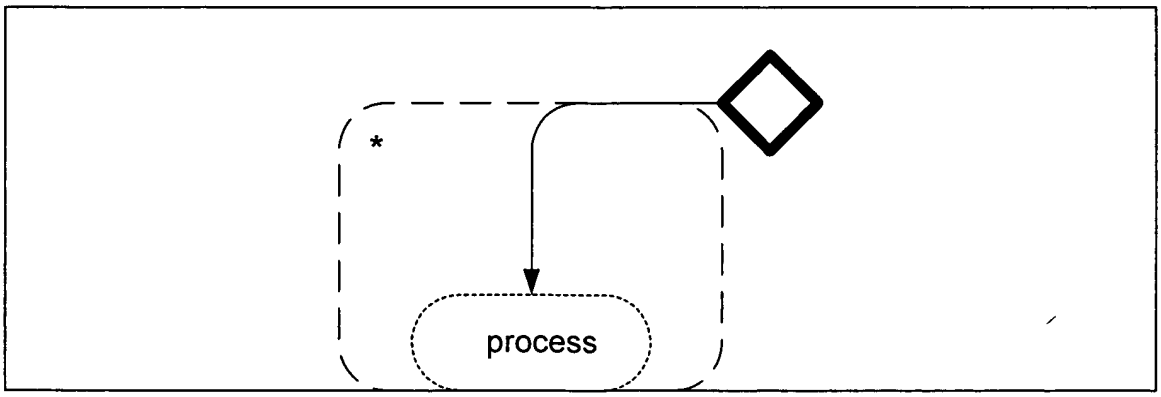

Fig. 20c

Example

```
<schedule name="loopExample">

<header>
<portList>
   <port name="p"/>
</portList>
<messageList>
   <message name-"mTrue"/>
   <message name="m"/>
</messageList>
</header>

<switch>
   <branch>
      <case>
         <ruleRef location="test"/>
         <msgRef location="mTrue"/>
         <msgRef location="m"/>
      </case>
      <sequence>
         <sink>
            <portRef location="p"/>
            <msgRef location="m"/>
         </sink>
         <call>
            <scheduleRef location="loopExample"/>
            <portRef location="p"/>
         </call>
```

Fig. 20d

Map (EBNF)		
map	: : =	assignmentList process contextRef?
assignmentList	: : =	*assignmentList* assignment*
assignment	: : =	*assignment* messageRef portRef

Fig. 21a

Map (XML)

```
< ! ELEMENT map ( (zero | sequence | switch | copy |
   partition | connect | cut), assignmentList, contextRef?)>

< ! ELEMENT assignmentList (assignment*)>

< ! ELEMENT assignment (messageRef, portRef)>
```

Fig. 21b

Example

```
<map>
   <assignmentList>
      <assignment>
         <messageRef location="m0"/>
         <portRef location="p1"/>
      </assignment>
   </assignmentList>
   <sequence>
      <sink>
         <portRef location="p0"/>
         <messageRef location="m0"/>
      </sink>
      <source>
         <portRef location="p1"/>
         <message location="m1"/>
      </source>
   </sequence>
</map>
```

Fig. 21c

Copy (EBNF)	
copy	: : = copy process contextRef?

Fig. 22a

Copy (XML)
< ! ELEMENT copy ((zero | sequence | switch | map | copy | partition | connect | cut), contextRef?)>

Fig. 22b

Partition (EBNF)	
partition	: : = process* contextRef?

Fig. 23a

Partition (XML)
< ! ELEMENT partition ((zero | sequence | switch | map | copy | partition | connect | cut)*, contextRef?)>

Fig. 23b

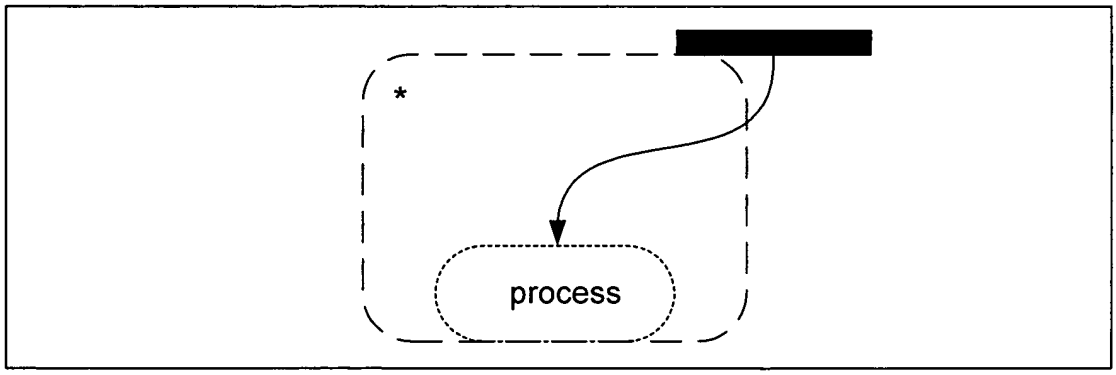

Fig. 23c

Connect (EBNF)		
connect	: : =	process process connectionList contextRef?
connectionList	: : =	*connectionList* portRef PortRef

Fig. 24a

Connect (XML)

```
< ! ELEMENT connect ( (zero | sequence | switch | map | copy |
      partition | connect | cut), (zero | sequence | switch | map |
      copy | partition | connect | cut), connectionList, contextRef?)>

< ! ELEMENT connectionList (connection*)>

< ! ELEMENT connection (portRef, portRef)>
```

Fig. 24b

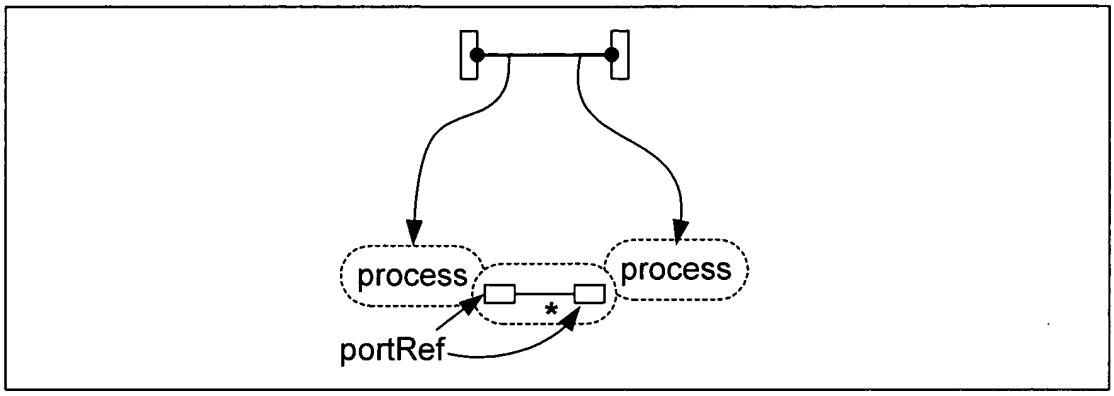

Fig. 24c

Cut (EBNF)		
cut	: : =	process process process contextRef?

Fig. 25a

Cut (XML)

```
< ! ELEMENT cut ( (zero | sequence | switch | map | copy |
      partition | connect | cut), (zero | sequence | switch | map |
      copy | partition | connect | cut), (zero | sequence | switch |
      map | copy | partition | connect | cut), contextRef?)>
```

Fig. 25b

```
<map>
  <cut>
    <partition>
      <sequence>
        <sink> <portRef location="x"/> <messageRef location="y"/> </
sink>
      </sequence>
      <sequence>
        <source> <portRef location="u"/> <messageRef location-"y"/> </
source>
      </sequence>
    </partition>
    <partition>
      <sequence>
        <sink> <portRef location="u"/> messageRef location="y"/> </
sink>
      </sequence>
      <sequence>
        <source> <portRef location="z"/> <messageRef location="w"/> </
source>                    </sequence>
    </partition
    <sequence>
      <sink> <portRef location="u"/> <messageRef location="v"/> </sink>
    </sequence>
  </cut>
  <assignmentList>
    <assignment>
      <messageRef location="y"/> <portRef location="z"/>
    </assignment>
  </assignmentList>
</map>
```

Fig. 26a

```
<connect>
   <sequence>
      <sink> <portRef loction="x"/> <messageRef location="y"/> </sink>
   </sequence>
   <sequence>
      <source> <portRef location="z"/> <messageRef location="w"/> </
source>
   </sequence>
   <connectionList>
      <conection>
         <portRef location="x"/> <portRef location="z"/>
      </connection>
   </connectionList>
</connect?
```

Fig. 26b

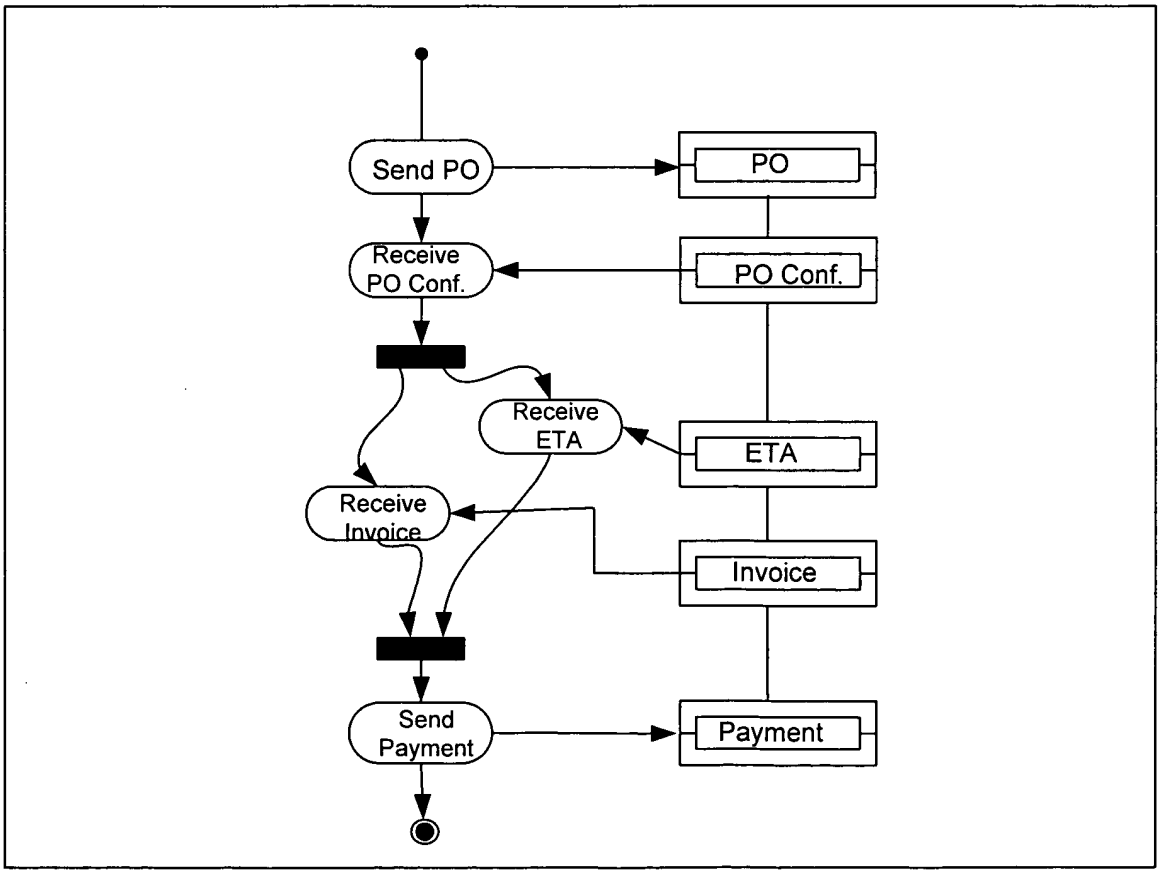

Fig. 27a

```
01    <schedule name="customer">
02
03    <header>
04       <portList>
05          <port name="pSendPO"/>
06          <port name="pReceiveConf"/>
07          <port name="pReceiveETA"/>
08          <port name="pReceiveInvoice"/>
09          <port name="pSendPayment"/>
10       </portList>
11       <messageList>
12          <message name="mPO"/>
13          <message name="mConf"/>
14          <message name="mETA"/>
15          <message name="mInvoice"/>
16          <message name="mPayment"/>
17       </messageList>
18    </header>
19
20    <sequence>
21       <source> <portRef location="pSendPO"/>
22                <messageRef location="mPO"/> </source>
23       <sink> <portRef location="pReceiveConf"/>
24              <messageRef location="mConf"/> </sink>
25    <task>
26       <sink> <portRef location-"pReceiveETA"/>
27              <messageRef location="mETA"/> </sink>
28       <sink> <portRef location="pReceiveInvoice"/>
29              <messageRef location-"mInvoice"/> </sink>
30    </task>
31    <source> <portRef location="pSendPayment"/>
32              <message location="mPayment"/> </source>
33    </sequence>
34
35    </schedule>
```

Fig. 27b

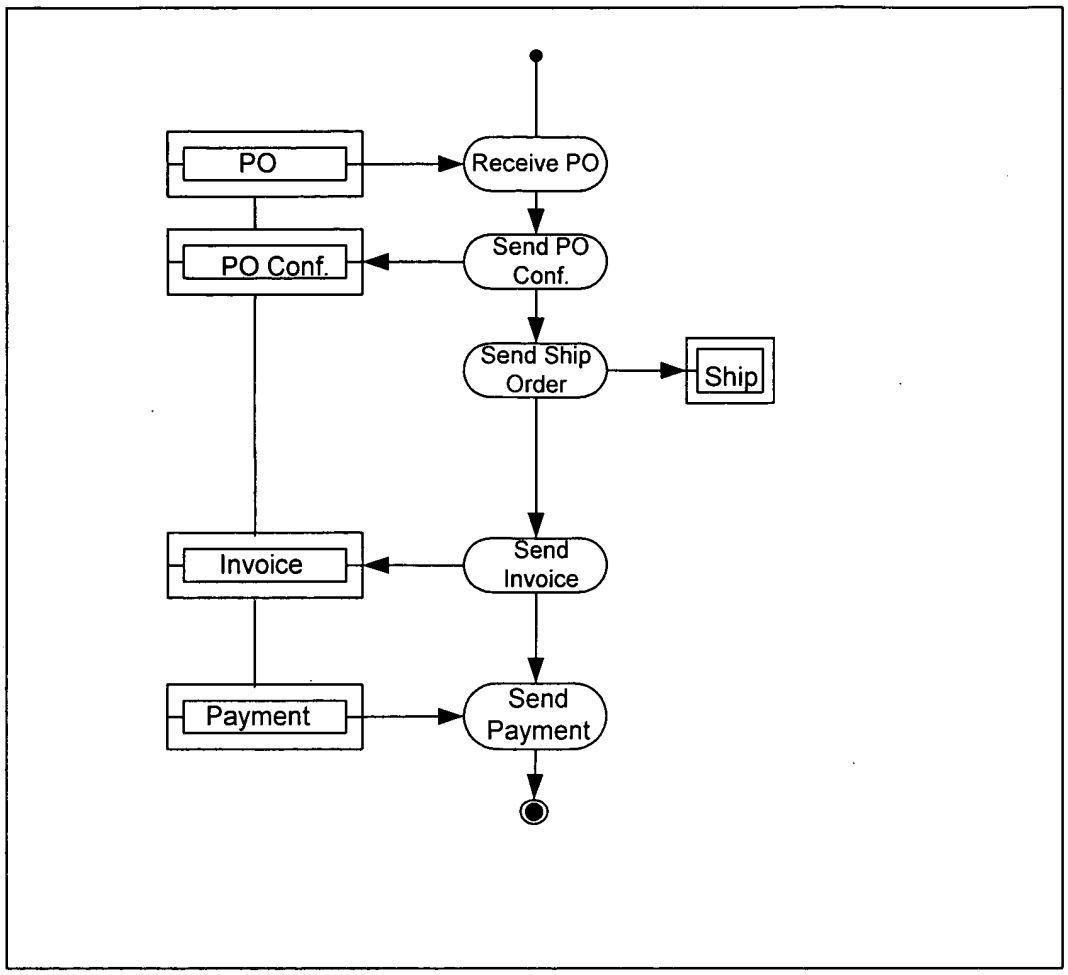
PO
Receive PO
PO Conf.
Send PO Conf.
Send Ship Order
Ship
Invoice
Send Invoice
Payment
Send Payment

Fig. 28a

```
01    <schedule name="supplier">
02
03    <header>
04       <portList>
05          <port name="pReceivePO"/>
06          <port name="pSendconf"/>
07          <port name="pSendShip"/>
08          <port name="pSendInvoice"/>
09          <port name="pReceivePayment"/>
10       </portList>
11       <messageList>
12          <message name="mPO"/>
13          <message name="mConf"/>
14          <message name="mShip"/>
15          <message name="mInvoice"/>
16          <message name="mPayment"/>
17       </messageList>
18    </header>
19
20    <sequence>
21       <sink> <portRef location="pReceivePO"/>
22                <messageRef location="mPO"/> </sink>
23       <source> <portRef location="pSendConf"/>
24              <messageRef location="mConf"/> </source>
25       <source> <portRef location="pSendShip"/>
26              <messageRef location="mShip"/> </source>
27       <source> <portRef location="pSendInvoice"/>
28              <messageRef location="mInvoice"/> </source>
29       <sink> <portRef location="pReceivePayment"/>
30                <messageRef location="mPayment"/> </sink>
31    </sequence>
32
33    </schedule?
```

Fig. 28b

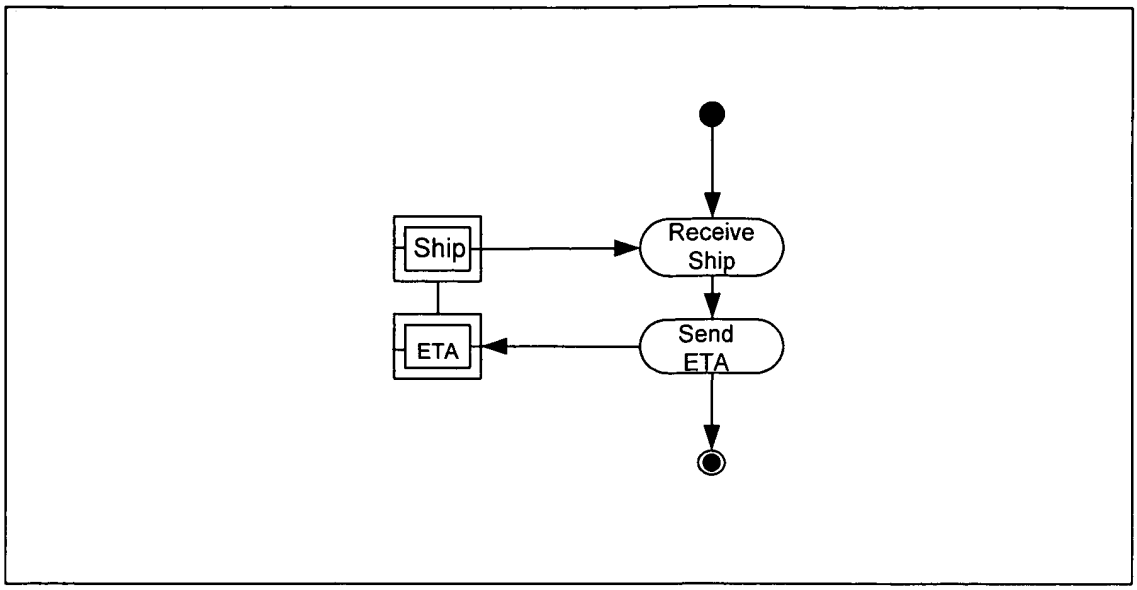

Fig. 29a

```
01    <schedule name="shipper">
02
03    <header>
04       <portList>
05          <port name="pReceiveShip"/>
06          <port name="pSendETA"/>
07       </portList>
08       <messageList>
09          <message name="mShip"/>
10          <message name="mETA"/>
11       <messageList>
12    </header>
13
14    <sequence>
15       <sink> <portRef location="pReceiveShip"/>
16              <messageRef location="mShip"/> </sink>
17       <source> <portRef location="pSendETA"/>
18              <messageRef location="mETZ"/> </source>
19    </sequence>
20
21    </schedule>
```

Fig. 29b

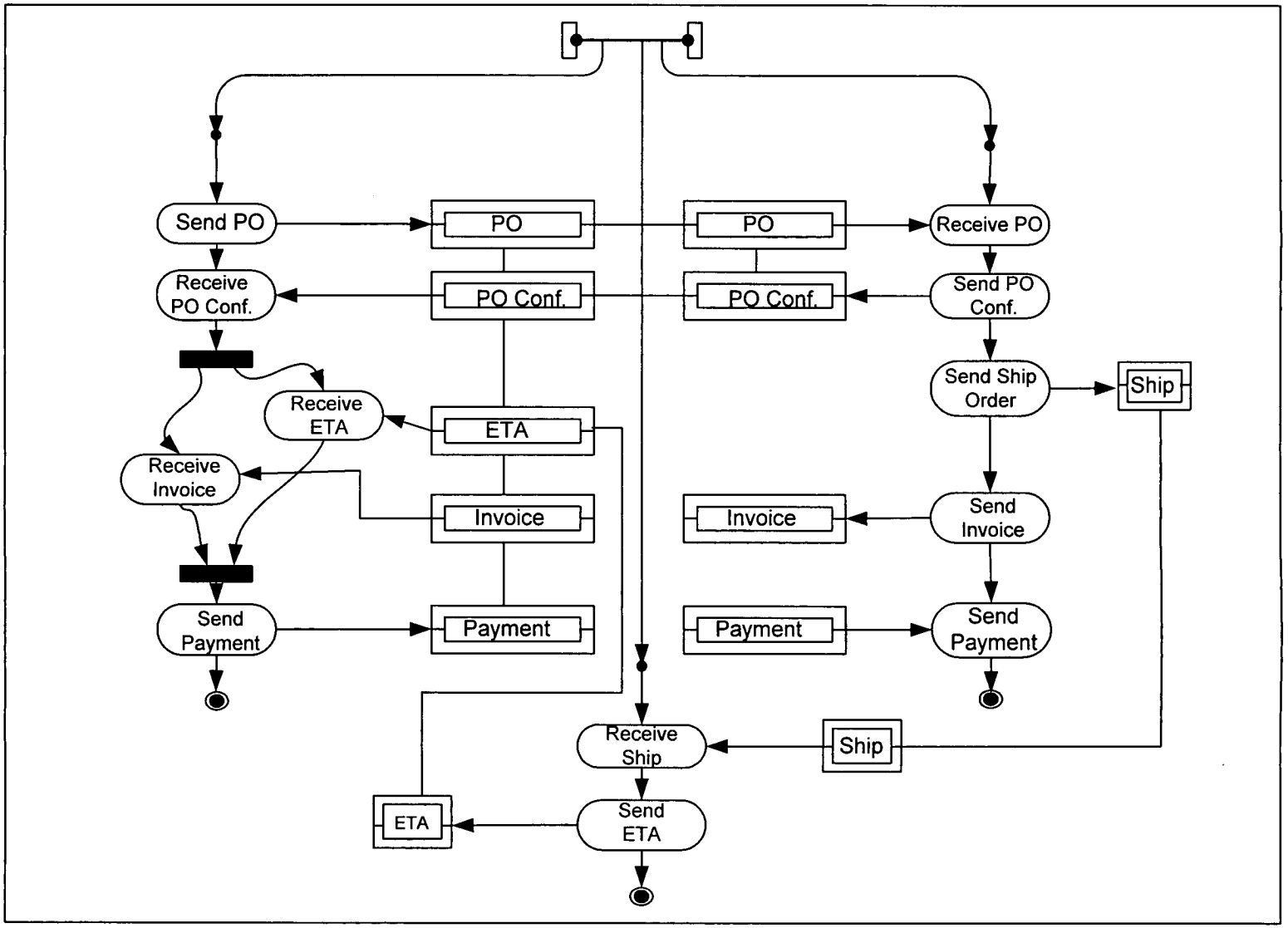
Send PO
PO
PO
Receive PO
Receive PO Conf.
PO Conf.
PO Conf.
Send PO Conf.
Send Ship Order
Ship
Receive ETA
ETA
Receive Invoice
Invoice
Invoice
Send Invoice
Send Payment
Payment
Payment
Send Payment
Receive Ship
Ship
ETA
Send ETA

Fig. 30a

```
01    <schedule name="customerSupplier">
02
03    <header>
04      <portList>
05        <port name="pCustomerPO"/>
06        <port name="pSupplierPO"/>
07        <port name="pCustomerPOConf"/>
08        <port name="pSupplierPOConf"/>
09        <port name="pCustomerETA"/>
10        <port name="pSupplierETA"/>
11        <port name="pCustomerInvoice"/>
12        <port name="pSupplierInvoice"/>
13        <port name="pCustomerPayment"/>
14        <port name="pSupplierPayment"/>
15        <port name="pSupplierShip"/>
16        <port name="pShipperShip"/>
17      </portList>
18    </header>
19
20    <connect>
21      <sequence>
22        <call>  <schedRef location="customer"/>
23                <portRef location="pCustomerPO"/>
24                <portRef location="pCustomerPOConf"/>
25                <portRef location="pCustomerETA"/>
26                <portRef location="pCustomerInvoice"/>
27                <portRef location="pCustomerPayment"/> </call>
28      </sequence>
29      <connect>
30        <sequence>
31          <call>  <schedRef location="supplier"/>
32                  <portRef location="pSupplierPO"/>
33                  <portRef location="pSupplierPOConf"/>
34                  <portRef location="pSupplierShip"/>
35                  <portRef location="pSupplierInvoice"/>
36                  <portRef location="pSupplierPayment"/> </call>
37        </sequence>
38        <sequence>
39          <call>  <schedRef location="shipper"/>
40                  <portRef location="pShipperShip"/>
41                  <portRef location="pSupplierETA"/> </call>
42        </sequence>
```

Fig. 30b

```
43      <connectionList>
44        <connection>  <portRef  location="pSupplierShip"/>
45                      <portRef  location="pShipperShip"/>  <connection>
46      </connectionList>
47    </connect>
48    <connectionList>
49        <connection>  <portRef  location="pCustomerPO"/>
50                       <portRef  location="pSupplierPO"/>  </connection>
51        <connection> <portRef  location="pCustomerPOConf"/>
52                      <portRef  location="pSupplierPOConf"/>  </connection>
53        <connection> <portRef  location="pCustomerETZ"/>
54                      <portRef  location="pSupplierETA"/>  </connection>
55        <connection> <portRef  location="PCustomerInvoice"/>
56                      <portRef  location="pSupplierInvoice"/>  </connection>
57        <connection> <portRef  location="pCustomerPayment"/>
58                      <portRef  location="pSupplierPayment"/>  </connection>
59      </connectionList>
60    </connect>
61
62    </schedule>
```

Fig. 30c

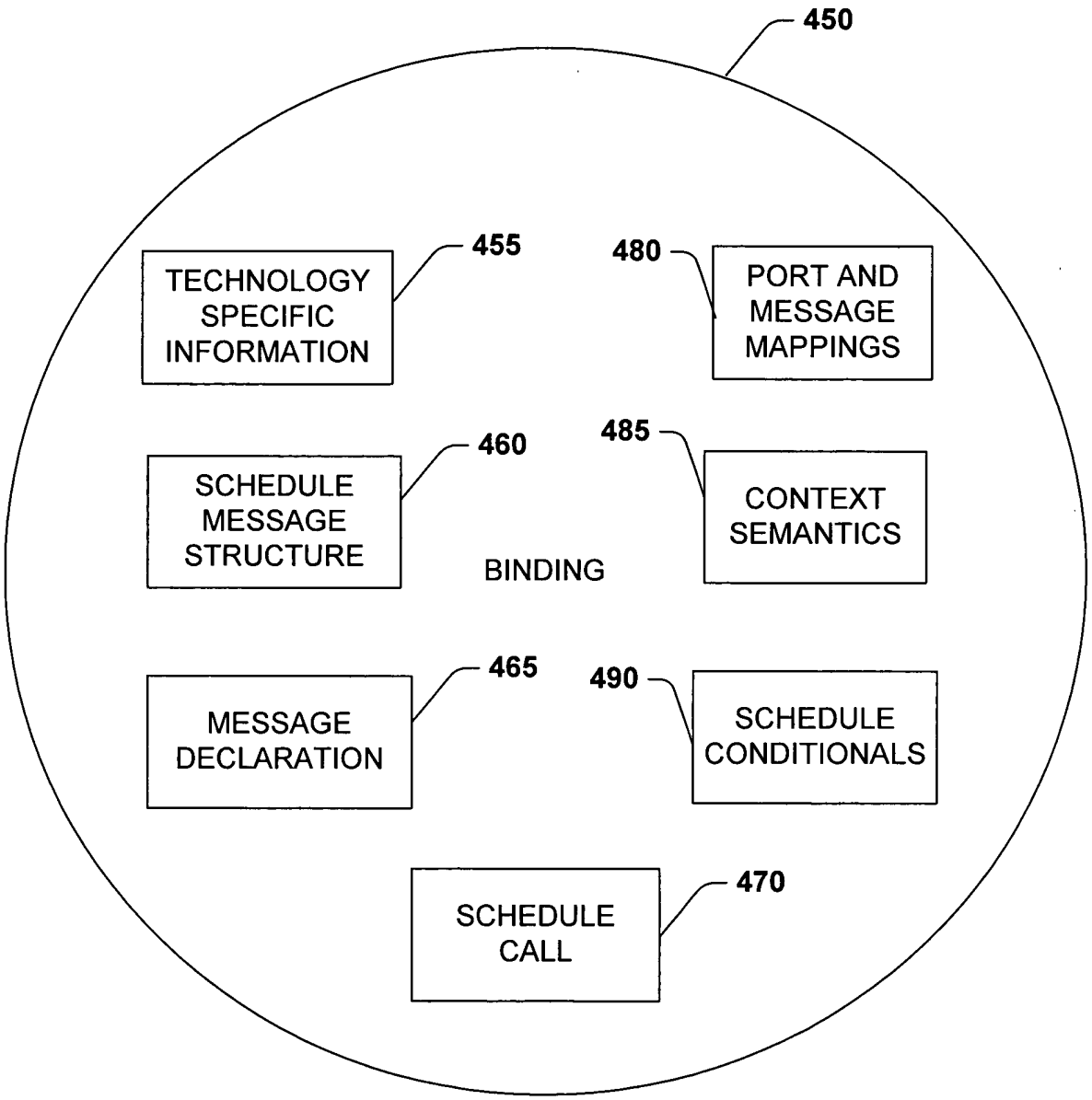
450
TECHNOLOGY SPECIFIC INFORMATION
455
480
PORT AND MESSAGE MAPPINGS
SCHEDULE MESSAGE STRUCTURE
460
485
CONTEXT SEMANTICS
BINDING
MESSAGE DECLARATION
465
490
SCHEDULE CONDITIONALS
SCHEDULE CALL
470

Fig. 31

BANG		
binding	: : =	scheduleRef translationHeaderList? schemaList? messageDeclList portBindindingList contextBindingList? ruleBindindingList? callBindingList?
translationHeaderList	: : =	translationHeader*
schemaList	: : =	schema*
messageDeclList	: : =	messageDecl*
messageDecl	: : =	messageRef messageTypeRef
portBindingList	: : =	portBinding*
portBinding	: : =	portRef portTranslation messageBindingList latency?
messageBindingList	: : =	messageBinding*
messageBinding	: : =	messageRef messageTranslation fieldBindingList latency?
fieldBindingList	: : =	fieldBinding*
fieldBinding	: : =	fieldRef from? provide? require? portRef?
contextBindingList	: : =	contextBinding*
contextBinding	: : =	contextRef (retry backoff) ? timeout?
ruleBindingList	: : =	ruleBinding*
ruleBinding	: : =	ruleRef messageRef messageRef (all | project matchList)
callBindingList	: : =	callBinding*
callBinding	: : =	TBD

Fig. 32

Binding (EBNF)

```
binding          : : =   identity? name?
                         scheduleRef
                         translationHeaderList?
                         schemaList?
                         messageDeclList
                         portBindindingList
                         contextBindingList?
                         ruleBindindingList?
                         callBindingList?
```

Fig. 33a

Binding (XML)

```
<ElementType name="binding" content="eltOnly">
   <attribute    type="name"/>
   <attribute    type="identity"/>

   <group order="seq">
      <element type="scheduleRef"/>
      <element type="translationHeaderList" minOccurs="0" maxOccurs="1"/>
      <element type="schemaList" minOccurs="0" maxOccurs="1"/>
      <element type="messageDeclList"/>
      <element type="portBindingList"/>
      <element type="contextBindingList" minOccurs="0" maxOccurs="1"/>
      <element type="ruleBindingList" minOccurs="0" maxOccurs="1"/>
      <element type="callBindingList" minOccurs="0" maxOccurs="1"/>
   </group>
</ElementType>
```

Fig. 33b

translationHeader (EBNF)

```
translationHeaderList     : : =   translationHeader*
translationHeader         : : =   <technology specific content>
```

Fig. 34a

translationHeader (XML)

```
<ElementType name="translationHeaderList">
   <element type="translationHeader" minOccurs="0" maxOccurs="*"/>
</ElementType>

>ElementType name="translationHeader"/>
```

Fig. 34b

schema (EBNF)
schemaList : : = schema*

Fig. 35a

schema (XML)
<ElementType name="schemaList"/>

Fig. 35b

messageDecl (EBNF)		
messageDeclList	: : =	messageDecl*
messageDecl	: : =	messageDecl messageRef messageTypeRef
messageTypeRef	: : =	messageTypeRef URI

Fig. 36a

messageDecl (XML)

```
<ElementType name="messageDeclList">
   <element type="messageDecl" minOccurs="0" maxOccurs="*"/>
<.ElementType>

<ElementType  name="messageDecl"  content="eltOnly">
   <group order="seq">
      <element type="messageRef"/>
      <element type="messageTypeRef"/>
   </group>
</ElementType>

<ElementType name="messageTypeRef">
   <attribute    type="location"/>
</ElementType>
```

Fig. 36b

portBinding (EBNF)		
portBindingList	: : =	portBinding*
portBinding	: : =	*portBinding portRef* portTranslation messageBindingList latency?
portTranslation	: : =	< technology specific content>

Fig. 37a

portBinding (XML)

```
<ElementType  name="portBindingList">
   <element type="portBinding" minOccurs="0" maxOccurs="*"/>
</ElementType>

<ElementType  name="portBinding" content="eltOnly">
   <group order="seq">
      <element type="portRef"/>
      <element type="portTranslation"/>
      <element type="messageBindingList"/>
      <element type="latency" minOccurs="0" maxOcurs="1"/>
   </group>
</ElementType>

<ElementType name="portTranslation"/>
```

Fig. 37b

messageBinding (EBNF)

messageBindingList	: : =	messageBinding*
messageBinding	: : =	*messageBinding messageRef* messageTranslation fieldBindingList latency?
messageTranslation	: : =	*<technology specific content>*

Fig. 38a

messageBinding (XML)

```
<ElementType name="messageBindingList">
   <element type="messageBinding" minOccurs="0" maxOccurs-"*"/>
</ElementType>

<ElementType name="messageBinding" content="eltOnly">
   <group order="seq">
      <element type="messageRef"/>
      <element type="messageTranslation"/>
      <element type="fieldBindingList"/>
      <element type="latency" minOccurs="0" maxOccurs="1"/>
   </group>
</ElementType>

<ElementType name="messageTranslation"/>
```

Fig. 38b

fieldBinding (EBNF)

fieldBindingList	: : =	fieldBinding*
fieldBinding	: : =	*fieldBinding fieldRef* from? provide? require? portRef?
fieldRef	: : =	*fieldRef URI*
fieldTranslation	: : =	*<technology specific content>*

Fig 39a

fieldBinding (XML)

```
<ElementType name="fieldBindingList">
   <element type="fieldBinding" minOccurs="0" maxOccurs="*"/>
<ElementType>

<ElementType name="fieldBinding" content="eltOnly">
   <group order="seq">
      <element type="fieldRef"/>
      <element type="fieldTranslation"/>
      <element type="from" minOccurs="0" maxOccurs="1"/>
      <element type="provide" minOccurs="0" maxOccurs="1"/>
      <element type="require" minOccurs="0" maxOccurs="1"/>
      <element type="portRef" minOccurs="0" maxOccurs="1"/>
   </group>
<ElementType>

<ElementType name="fieldRef">
   <attribute    type="location"/>
</ElementType>

<ElementType name="fieldTranslation"/>
```

Fig. 39b

from (EBNF)

from	: : =	*from fieldRef*

Fig. 40a

from (XML)

```
<ElementType name="from">
  <element type="fieldRef"/>
<ElementType>
```

Fig. 40b

provide (EBNF)		
provide	: : =	*provide*

Fig. 41a

provide (XML)
<ElementType name="provide" content="empty"/>

Fig. 41b

require (EBNF)		
require	: : =	*require*

Fig. 42a

require (XML)
<ElementType name="require" content="empty"/>

Fig. 42b

latency (EBNF)		
latency	: : =	*latency* time

Fig. 43a

latency (XML)
<ElementType name="latency" dt: type="int"/>

Fig. 43b

contextBinding (EBNF)		
contextBindingList	: : =	contextBinding*
contextBinding	: : =	*contextBinding contextRef* (retry backoff) ? timeout?

Fig. 44a

contextBinding (XML)

```
<ElementType name="contextBindingList">
   <element type="contextBinding" minOccurs-"0" maxOccurs="*"/>
<ElementType>

<ElementType name="contextBinding" content="eltOnly">
   <group order="seq">
      <element type="contextRef"/>
      <group order="seq" minOccurs="0" maxOccurs="1">
        <element type="retry"/>
        <element type="backoff"/>
      </group>
      <element type="timeout"/>
   </group>
</ElementType>
```

Fig. 44b

retry (EBNF)

retry : : = *retry count*

Fig. 45a

retry (XML)

```
<ElementType name="retry" dt: type="int"/>
```

Fig. 45b

backoff (EBNF)

backoff : : = *backoff time*

Fig. 46a

backoff (XML)

```
<ElementType name="backoff" dt: type="int"/>
```

Fig. 46b

Timeout (EBNF)

timeout : : = *timeout time*

Fig. 47a

Timeout (XML)

```
<ElementType name="timeout" dt: type="int"/>
```

Fig. 47b

ruleBinding (EBNF)

ruleBindingList	: : =	ruleBinding*
ruleBinding	: : =	*ruleRef messageRef messageRef*
		(all | project matchList)
matchList	: : =	match*
match	: : =	*match fieldRef fieldRef*

Fig. 48a

ruleBinding (XML)

```
<ElementType name="ruleBindingList">
  <element type="ruleBinding" minOccurs="0" maxOccurs="*"/>
</ElementType>

<ElementType name="ruleBinding" content="eltOnly">
  <group order="seq">
    <element type="ruleRef"/>
    <element type="messageRef"/>
    <element type="messageRef"/>
    <group order="one">
    <element type="all"/>
    <group order="seq">
        <element type="project"/>
        <element type="matchList"/>
      </group>
    </group>
  </group>
</ElementType>

<ElementType name="all" content="empty"/>

<ElementType name="project" content="empty"/>

<ElementType name="matchList">
  <element type="match" minOccurs="0" maxOccurs="*"/>
</ElementType>

<ElementType name="match" content="eltOnly">
  <group order="seq">
    <element type="fieldRef"/>
    <element type="fieldRef"/>
  </group>
</ElementType>
```

Fig. 48b